■ ปัญญา เปรมปรีดิ์

หลักสูตรคอมพิวเตอร์ กับการพัฒนาทางด้านไอที

หลักสูตรนักคอมพิวเตอร์ ในสายตาของผม

แนวทางในการพัฒนา
นักคอมพิวเตอร์
ที่เหมาะสมให้แก่ประเทศได้

ช่วงนี้ (เม.ย. 2542) มีคนและสถาบันการศึกษาออกมาเสนอแนวคิดการแก้ปัญหาคนตกงานด้วยการสอนคอมพิวเตอร์กันมาก บางคนก็อ้างว่าศูนย์กีฬาที่รังสิตนั่นว่างอยู่ และมีเครื่องไมโครคอมพิวเตอร์อยู่เยอะแยะจึงควรเอามาใช้งาน บางคนก็อ้างว่า เมืองไทยนั้นขาดโปรแกรมเมอร์และผู้วิเคราะห์ระบบเป็นจำนวนมาก บางคนบอกว่า มาถึงยุคนี้ก็ต้องรีบประยุกต์คอมพิวเตอร์เข้ากับงานต่าง ๆ พวกหลังนี้อ้างว่า คำว่า "วิกฤต" ในภาษาจีนนั้นเป็นคำสองคำ คือ "คำว่าข้าวยากหมากแพง" และคำว่า "โอกาส" ดังนั้นเราจึงควรพลิกผันความยุ่งยากมาเป็นโอกาสในการพัฒนาประเทศ แต่บางคนนั้นเริ่มงานมาปีหนึ่งแล้ว กลุ่มนี้เปิดสอนความรู้ทางคอมพิวเตอร์ 3–4 วิชา คือการเขียนโปรแกรมด้วยภาษาวิชวลเบสิก การเขียนโปรแกรมด้วยภาษาจาวา การใช้โปรแกรมสำเร็จรูปพาวเวอร์บิลด์เดอร์ และการวิเคราะห์และออกแบบระบบ แล้วก็เตรียมที่จะเปิดวิชาที่เข้าลักษณะ "แอดวานซ์" ต่าง ๆ แต่ทั้งหมดนี้มุ่งเป้าไปที่คนที่ตกงานเพราะภาวะเศรษฐกิจซบเซา ซึ่งมีมากกว่า 2 ล้านคน พวกเขาคิดกันว่าจะสามารถเปิดสอนได้งวดละ 10,000 คน ค่าเล่าเรียนหัวละ 40,000 กว่าบาท และเพื่อเป็นการช่วยเหลือคนเหล่านี้ได้ยามยาก เขาจะมีเงินจากกรมพัฒนาฝีมือแรงงานมาให้กู้ 30,000 บาท โดยคิดดอกเบี้ยต่ำมาก คือปีละ 1 เปอร์เซ็นต์เท่านั้น โครงการนี้ดูแล้วจะอยู่ไปได้หลายปี

ครับ ผมเห็นว่าเป็นความคิดที่บ้าบอคอแตกที่สุด ผมได้เขียนบ่นไปแล้ว 1 ครั้ง ที่ผมเห็นว่าไม่เหมาะสมก็เพราะคนที่ตกงาน 2 ล้านคนนั้นส่วนใหญ่จะไม่มีพื้นฐานที่จะเรียนเป็นโปรแกรมเมอร์หรือผู้วิเคราะห์ระบบได้ เปิดสอนไปแล้วจะเรียนไม่รู้เรื่อง แล้วก็จะทิ้งเสียก่อนจบหลักสูตร คนที่เรียนจนจบได้นั้นก็มิใช่ว่าจะทำงานได้ทุกคน จะมีเพียงไม่ถึง 1 เปอร์เซ็นต์ที่จะพัฒนาตนเองจนสำเร็จในวิชาชีพอันนี้ได้ อีกประการหนึ่งคือ นักคอมพิวเตอร์ที่ประเทศไทยต้องการนั้น มันเป็นจำนวนเพียงไม่กี่พันคน แล้วจะมาผลิตกันปีละ 20,000 คน (2 งวดต่อปี หลักสูตรนี้ยาว 6 เดือน) ทีเดียวหรือ ผลิตกันมากมายไปทำไม ไม่ช้าไม่นานก็เกลื่อนถนน และประการสุดท้ายก็คือ การที่จะเป็นนักคอมพิวเตอร์ได้นั้น ผมเห็นว่ามันต้องใช้เวลา 18 ถึง 24 เดือน จะมาสอนกันอย่างลวก ๆ นั้นไม่ได้หรอก

จากคำบ่นของผม ก็เลยมีคนเขาอยากได้ "หลักสูตรนักคอมพิวเตอร์" ในแนวคิดของผม มีบางคนแวะไปคุยกับผมถึงที่บ้าน เขาชี้ให้เห็นว่าถ้ามีหลักสูตรที่ดีจริง เขาสามารถทำเงินได้ไม่น้อยเลย ตัวอย่างเช่น ถ้าเปิดสอนรุ่นละ 10 ห้องเรียน ห้องเรียนละ 50 คน ก็จะเป็นจำนวน 500 คน ค่าเล่าเรียนหัวละ 40,000 บาท ก็จะเป็นเงินถึง 20 ล้านบาท ถ้าหลักสูตรนี้ประสบความสำเร็จ ก็จะขายหลักสูตรและตำราประกอบการสอนให้คนอื่นเขาเอาไปทำต่อได้อีกหลายสตางค์ ทั้งนี้เพราะความต้องการนั้นสูงกว่า 5,000 คน และคงจะมีเพิ่มขึ้นไปเรื่อย ๆ ในอนาคต เรื่องนี้ผมได้แต่หัวเราะหึหึ ทั้งนี้เพราะผมแก่แล้ว ผมไม่สามารถมานั่งเขียนตำราและกำกับการสอนได้ ที่ผมพอจะช่วยได้ก็คือ จะเขียนเป็นบทความให้อ่านกัน ใครเห็นด้วยก็นำไปพัฒนาเอกสารประกอบแล้วเปิดสอนไป หลักสูตรของผมนี้ ถ้าทำได้เต็มตามแนวคิดของผม ค่าเล่าเรียนหัวละ 100,000 บาทก็คงไม่แพง ทั้งนี้เพราะความรู้จะแน่นเท่า ๆ กับเรียนปริญญาตรีเลยทีเดียว แต่จะเน้นเฉพาะจุดที่จะต้องนำไปใช้งานได้จริง ๆ และใช้เวลาถึง 2 ปี

โอเค เรามาดูกันว่าหลักสูตรของผมจะเป็นอย่างไร

หลักสูตรคอมพิวเตอร์กับการพัฒนาทางด้านไอที

แนวคิดอันแรกของผมก็คือ *จะต้องสร้างคนให้มีความรู้ทางคอมพิวเตอร์ให้รอบด้าน หรือคือ ทำงานได้ทั้งในหน้าที่เป็นโปรแกรมเมอร์ เป็นผู้วิเคราะห์และออกแบบระบบ และเป็นทั้งผู้ควบคุมโครงการได้* นี่จึงจะเรียกว่า "นักคอมพิวเตอร์" ที่แท้จริง เราจะสอนแค่เป็นโปรแกรมเมอร์อย่างเดียว หรือผู้วิเคราะห์และออกแบบระบบอย่างเดียวไม่ได้ งานพัฒนาและดูแลงานด้านคอมพิวเตอร์นี้ต้องทำงานเป็นทีม และต้องแบ่งงานกันทำในทุกขั้นตอน กล่าวคือ ในช่วงของการวิเคราะห์และออกแบบระบบนั้นทุกคนต้องไปสำรวจและเก็บข้อมูลด้วยกัน คนที่มีประสบการณ์สูงกว่าจะช่วยเป็นผู้ชี้ว่าต้องดูและเก็บอะไรมาบ้าง ส่วนคนที่มีประสบการณ์น้อยก็เป็นลูกมือไปก่อน และเมื่อถึงขั้นลงมือเขียนโปรแกรม ทุกคนก็ต้องรับงานไปทำ คนมือใหม่สุดก็จะทำโปรแกรมง่าย ๆ คนมือมีประสบการณ์พอสมควรแล้วก็จะทำโปรแกรมยาก ๆ คนที่มีประสบการณ์สูงสุดจะเป็นคนควบคุมโครงการ เป็นคนเจรจากับยูสเซอร์เมื่อมีข้อขัดข้องหรือข้อสงสัย และต้องเก็บรวบรวมเอกสารต่าง ๆ ไว้เพื่อทำเอกสารประกอบระบบ

แนวคิดอันที่ 2 ของผมก็คือ *จะเรียนรู้เรื่องใดก็จะต้องมีความรู้พื้นฐานที่แน่นพอที่จะเข้าใจ และอธิบายให้คนอื่นฟังได้* อย่างเช่น จะเข้าใจเรื่องของเมมโมรีประเภท RAM ก็จะต้องรู้การทำงานของทรานซิสเตอร์ รู้จักวงจรไฟฟ้าที่จะเก็บข้อมูลแต่ละบิต (flip-flop) รู้วิวัฒนาการจากวงจรที่ใช้คอมโปเนนต์เป็นชิ้น ๆ ไปสู่วงจรรวม (Integrated Circuit) ความรู้พื้นฐานเหล่านี้ไม่ถึงขั้นจะต้องผลิตเองหรือทำเองได้ แต่ก็ต้องเพียงพอสำหรับการอธิบายให้คนทั่ว ๆ ไปเข้าใจได้ มิใช่เป็นการท่องจำ หากจะต้องขุดลึกลงไปกว่านั้นก็ต้องรู้ว่าจะไปหารายละเอียดได้ที่ไหน ความรู้พื้นฐานสำหรับการทำความเข้าใจด้านฮาร์ดแวร์คือ วิชาฟิสิกส์และไฟฟ้าอิเล็กทรอนิกส์ ความรู้พื้นฐานสำหรับทำความเข้าใจด้านซอฟต์แวร์คือ คณิตศาสตร์และตรรกวิทยา ส่วนความรู้พื้นฐานที่จะใช้ทำความเข้าใจกับเรื่องระบบงานคือ สถิติและบริหารธุรกิจ นี่เป็นการมองอย่างกว้าง ๆ แต่เมื่อลงรายละเอียดในชิ้นส่วนอันใดอันหนึ่งของฮาร์ดแวร์ หรือซอฟต์แวร์ หรือระบบงาน ก็จะลงไปถึงทฤษฎีและหลักการพื้นฐานในวิชาต่าง ๆ ที่กล่าวมาจำนวนหนึ่งเท่านั้น นั่นคือ ไม่จำเป็นจะต้องเชี่ยวชาญในวิชาฟิสิกส์ ไฟฟ้า คณิตศาสตร์ ตรรกวิทยา สถิติ หรือบริหารศาสตร์ จนถึงขั้นเป็นผู้เชี่ยวชาญ ความรู้พื้นฐานเหล่านี้จะเชื่อมโยง หรือส่งเสริมให้เข้าใจโลกของคนที่อยู่ในวิชาชีพคอมพิวเตอร์ได้เป็นอย่างดี

แนวคิดอันที่ 3 ของผมก็คือ *จะเลือกหนทางปฏิบัติใดจะต้องมีความรู้พอที่จะอธิบายได้ว่าวิธีการนั้น ๆ ดีอย่างไร* นี่หมายถึงว่าจะต้องเปรียบเทียบข้อดีข้อเสีย หรือประมาณเป็นผลประโยชน์เป็นตัวเงินออกมาได้ ข้อกำหนดอันนี้ทำให้ต้องรู้ว่าอุปกรณ์และเทคนิคต่าง ๆ นั้นมีราคาเท่าใด จะต้องใช้เป็นจำนวนเท่าใด ระบบจะทำงานด้วยความเร็วประมาณเท่าใด ต้องใช้แรงงานในการพัฒนาเท่าใด ต้องใช้แรงงานเพื่อปฏิบัติงานประจำวันเท่าใด รวมเป็นค่าใช้จ่ายต่อปีหรือต่อครั้งเท่าใด และจะให้ผลงานคุ้มกับค่าใช้จ่ายเหล่านี้หรือไม่ เรื่องนี้จะพูดอีกอย่างหนึ่งก็ได้ คือต้องคิดในแง่เศรษฐศาสตร์ตลอดเวลา หรือต้องพัฒนาเขาให้เป็นคนที่มีหัวทางธุรกิจให้ได้ งานคอมพิวเตอร์ไม่ใช่งานศิลป์ และก็ไม่ใช่งานค้นคว้าวิจัยเพื่อความรู้ทางวิทยาศาสตร์ งานคอมพิวเตอร์เป็นงานประยุกต์ ทำเพื่อให้เกิดการผลิตหรือการใช้ที่คุ้มทุน

แนวคิดอันที่ 4 ของผมก็คือ *จะทำงานได้ต้องมีภาคปฏิบัติจนรู้แจ้งเห็นจริง* งานด้านคอมพิวเตอร์นี้คิดง่าย แต่ทำยาก โดยเฉพาะอย่างยิ่ง เรื่องของการเขียนโปรแกรม เวลาคิดวางเค้าโครงของโปรแกรมนั้นรวดเร็วมาก แต่เวลาเขียนออกมาเป็นตัวโปรแกรมจริง ๆ แล้วมักจะทำงานไม่ตรงกับที่คิด เรามักจะหลงลืม เรามักจะคิดได้ไม่รอบคอบ ซึ่งทำให้ต้องเสียเวลาและค้นหาที่ผิดเป็นเวลาอันยาวนาน ประสบการณ์จะช่วยให้หาที่ผิดได้เร็วขึ้น ประสบการณ์จะช่วยบอกว่าอย่าคิดแบบนั้นหรือแบบนี้ เพราะมันเคยผิดพลาดมาแล้วเยอะ ประสบการณ์จะช่วยให้เลือกวิธีคิดและวิธีทำที่ได้ผลเร็ว

แนวคิดอันสุดท้ายของผมก็คือ *เรียนรู้แต่ที่จำเป็นต่อการใช้งานจริง ๆ เสียก่อน เมื่อออกไปปฏิบัติงานแล้วค่อยไปแสวงหาอย่างอื่นมาช่วยขัดเกลาให้ความคิดแหลมคมยิ่งขึ้น* ยกตัวอย่างเช่น นักคอมพิวเตอร์ผู้เริ่มต้นนั้นไม่ต้องรู้วิธีสร้างคอมไพเลอร์ (Compiler Construction) ไม่ต้องรู้ทฤษฎีเครื่องจักรกลอัตโนมัติ (Automata Theory) ไม่ต้องรู้ทฤษฎีข้อสนเทศ (Information Theory) .... ฯลฯ ซึ่งก็หมายถึงว่า เราสามารถตัดวิชาออกไปจากหลักสูตรปริญญาตรีทางคอมพิวเตอร์ที่ใช้สอนกันในมหาวิทยาลัยได้กว่าครึ่ง วิชาความรู้เหล่านี้ควรไปเรียนรู้เอาทีหลัง จะโดยวิธีอ่านตำรา หรือจะไปเรียนภาคค่ำของมหาวิทยาลัยก็ได้ วิชาเหล่านี้เหมาะสำหรับคนที่ต้องยึดอาชีพเป็นนักคอมพิวเตอร์จริง ๆ แล้วเท่านั้น ถ้าท่านยังไม่ผ่านหลักสูตรของผม ท่านอย่าหวังว่าจะเรียนวิชาที่กล่าวมาข้างบนนี้เข้าใจได้

โอเค จากแนวคิดต่าง ๆ นี้ ผมบอกได้ว่า ความรู้ที่สำคัญของนักคอมพิวเตอร์เบื้องต้นนั้นมีแค่ 5 วิชาเท่านั้น ซึ่งได้แก่

1. ความรู้ด้านฮาร์ดแวร์

หลักสูตรคอมพิวเตอร์กับการพัฒนาทางด้านไอที

2. ความรู้ด้านซิสเต็มส์ซอฟต์แวร์

3. ความรู้ด้านออกแบบโปรแกรมและเขียนโปรแกรม

4. ความรู้ด้านออกแบบระบบและทำเอกสารประกอบระบบ

5. ความรู้ด้านการควบคุมโครงการและติดตั้งใช้งาน

นอกจาก 5 วิชาหลักนี้แล้ว นักศึกษาควรทบทวนความรู้พื้นฐานที่จะใช้ทำความเข้าใจในวิชาหลัก ซึ่งผมขอจับเอามารวมกันเป็น 3 วิชา ดังนี้

6. ทบทวนความรู้ด้านฟิสิกส์และไฟฟ้า

7. ทบทวนความรู้ด้านคณิตศาสตร์และตรรกวิทยา

8. ทบทวนความรู้ด้านสถิติและบริหารธุรกิจ

หลังจากแบ่งเป็นหัวข้อวิชาแล้ว ก็จะมีคำถามว่าแต่ละวิชาควรใช้ชั่วโมงเรียนและชั่วโมงฝึกฝนสักเท่าใด? และจะต้องจัดลำดับก่อนหลังกันอย่างไร?

ในเรื่องลำดับการสอนนั้น วิชาประเภททบทวนความรู้พื้นฐานต้องมาก่อนแน่ ๆ และทั้ง 3 วิชาเรียนขนานกันไปได้เลย ผมคิดว่าทั้ง 3 วิชานี้ควรจะใช้เวลาไม่เกิน 2 เดือน นานกว่านี้จะเบื่อและลืมหมด หลังจากปูพื้นฐานเสร็จแล้วก็น่าจะเริ่มวิชาความรู้ด้านฮาร์ดแวร์ เมื่อสอนฮาร์ดแวร์ไปได้ประมาณครึ่งทาง ซึ่งจะครอบคลุมอุปกรณ์หลัก ๆ ของเครื่องคอมพิวเตอร์แบบเมนเฟรม มินิคอม–พิวเตอร์ และไมโครคอมพิวเตอร์ไปแล้ว ก็น่าจะเริ่มวิชาความรู้ด้านซิสเต็มส์ซอฟต์แวร์ และเมื่อสอนซิสเต็มส์ซอฟต์แวร์ไปได้ประมาณครึ่งทางแล้วก็ควรเริ่มวิชาความรู้ด้านออกแบบโปรแกรมและเขียนโปรแกรม ถึงตอนนี้ความรู้ด้านฮาร์ดแวร์น่าจะจบไปแล้ว เมื่อสอนความรู้ด้านโปรแกรมไปได้ครึ่งทางก็น่าจะเริ่มวิชาความรู้ด้านออกแบบระบบและด้านควบคุมโครงการ 3 วิชาหลังนี้น่าจะจบได้พร้อม ๆ กัน ท่านผู้อ่านควรสังเกตว่า วิชาที่ 1. ถึง 5. นั้นจะมีเนื้อหาและเวลาเรียนไม่เท่ากัน วิชาด้านออกแบบและเขียนโปรแกรมนั้นจะใช้เวลานานที่สุด

ที่กล่าวมานั้นเป็นการจัดตารางสอนชนิดที่จะทำให้ความรู้ในวิชาหนึ่งไปช่วยในการเรียนรู้ของวิชาข้างเคียงได้ แต่นักศึกษาจะเรียนหนัก และเรียนหลายวิชาในช่วงตอนกลาง ๆ ของหลักสูตร โอกาสที่จะมีคนถอยหนี (Drop–out) มีพอสมควร สำหรับคนที่ชอบเรียนทีละเรื่อง และมีความจำดี (ตรงกับอุปนิสัยของคนไทยเรา) ผมแนะนำว่าวิชาหลัก 5 อย่างนั้นอาจสอนตามลำดับกันไป นั่นคือ อัดความรู้ด้านฮาร์ดแวร์เข้าไปก่อน เสร็จแล้วอัดความรู้ด้านซิสเต็มส์ซอฟต์แวร์ แล้วจึงให้รู้จักเขียนโปรแกรม เมื่อจบด้านเขียนโปรแกรมแล้วก็มาเรียนเป็นผู้วิเคราะห์และออกแบบระบบ หลังจากนั้นก็มาเรียนเป็นผู้ควบคุมโครงการ การจัดตารางสอนแบบที่ 2 นี้สะดวกกับผู้สอนด้วย คือผู้สอนสามารถผ่อนสั้นผ่อนยาวด้วยตนเองได้ ถ้าอัดได้เร็วนักศึกษาก็จะจบได้เร็วขึ้น ท่านชอบแบบไหนก็เอาแบบนั้นไปใช้ ผมคิดว่าจะให้ผลไม่แตกต่างกันนัก แบบที่ 2 นี้อาจลดความสับสนของนักศึกษาได้มาก สำหรับจำนวนชั่วโมงการสอนของ 5 วิชาหลักนี้ผมจะประมาณให้ในภายหลัง ในตอนนี้เรามาพูดถึงหัวข้อย่อยที่จะจัดเข้าไว้ในแต่ละวิชาจะดีกว่า หัวข้อย่อยในแต่ละวิชานั้นผมจะเขียนตามลำดับไปเรื่อย ๆ ผมจะใช้เครื่องหมายเซมิคอล่อน (;) เป็นตัวแบ่งระหว่างหัวข้อย่อยเหล่านี้ และถ้ามีข้อคิดหรือข้อสังเกตที่ต้องย้ำเป็นพิเศษก็จะเขียนเอาไว้ในวงเล็บ

หัวข้อย่อยในวิชาต่าง ๆ ควรเป็นดังนี้

1. ความรู้ด้านฮาร์ดแวร์ : ประวัติความเป็นมาของเครื่องเมนเฟรม (แนวคิดในการควบคุมเครื่องทอผ้าด้วยบัตรเจาะรู, เครื่อง ENIAC, เครื่องไอบีเอ็ม, เครื่องยี่ห้ออื่น ๆ, รุ่นและยี่ห้อที่ควรรู้จัก, ราคาโดยประมาณ, วิธีการจัดหา) ; ระบบแสดงข้อมูลแบบไบนารี (สวิตช์เปิดปิด, ขั้วแม่เหล็ก, วงจรไฟฟ้า flip–flop, ข้อดีของระบบไบนารี) ; โครงสร้างของเครื่องเมนเฟรม (CPU, Console, Memory Unit, Arithmatic Unit, Control Unit, I/O Unit) ; หน่วยวัดที่เกี่ยวกับคอมพิวเตอร์ (บิต, ไบต์, กิโลไบต์, เมกะไบต์, กิกะไบต์, วินาที, มิลลิวินาที, ไมโครวินาที, นาโนวินาที) ; อุปกรณ์ที่ใช้กับเมนเฟรม (จานแม่เหล็กถอดได้, จานแม่เหล็กถอดไม่ได้, เทปม้วน, เทปตลับ, ดิสก์เกตต์, เครื่องพิมพ์, ดิสเพลย์เทอร์มินัล, พรินเตอร์เทอร์มินัล, วิธีการบันทึกและอ่านข้อมูล, ความเร็วในการทำงาน, ขนาดความจุ, ราคาโดยประมาณ, วิธีการจัดหา) ; ตัวควบคุมอุปกรณ์ที่ใช้กับเมนเฟรม, (ตัวควบคุมจานแม่เหล็ก, ตัวควบคุมเทปแม่เหล็ก, การจัดตัวควบคุมกับตัวไดรฟ์เพื่อให้สามารถทำงานขนานกันได้หลายทาง, ความเร็ว, ความสามารถ, ราคา) ; ระบบสื่อสารของเมนเฟรม (ดาต้าคอมโปรเซสเซอร์, โมเด็ม, สายสื่อสาร, ตัวควบคุมเทอร์มินัล, ดิสเพลย์เทอร์มินัล, พาสบุ๊คพรินเตอร์, เทอร์มินัลพรินเตอร์, อีมูเลเตอร์กับไมโครคอมพิวเตอร์, วิธีการรับส่งข้อมูล, ความเร็วในการรับส่ง, ความสามารถ (จำนวนช่องสัญญาณ, การขยาย), ราคา, ค่าใช้จ่ายด้านสายสื่อสาร) ; ประวัติความเป็นมาของมินิคอมพิวเตอร์ (ความต้องการของหน่วยย่อย, (โรงงาน, ฝ่ายผลิต, สำนักงานภาค), ลักษณะภายนอก, ลักษณะภายใน, รุ่นและยี่ห้อที่ควรรู้จัก, ราคาโดยประมาณ, วิธีการจัดหา) ; โครงสร้างเครื่องมินิคอมพิวเตอร์ (Bus, Memory Unit, Arithmatic Unit, Control Unit, I/O Unit) ; อุปกรณ์ที่ใช้กับมินิคอมพิวเตอร์ (จานแม่เหล็ก

หลักสูตรคอมพิวเตอร์กับการพัฒนาทางด้านไอที

ถอดได้, จานแม่เหล็กถอดไม่ได้, เทปม้วน, เทปตลับ, ดิสก์เกตต์, เครื่องพิมพ์, ดิสเพลย์เทอร์มินัล, กราฟิกเทอร์มินัล, ความเร็ว, ขนาดความจุ, ราคาโดยประมาณ, วิธีการจัดหา) ; ตัวควบคุมอุปกรณ์ (ตัวควบคุมชนิดอยู่บนบอร์ดร่วมกับซีพียู, ตัวควบคุมชนิดอยู่ภายนอก) ; ระบบสื่อสารของมินิคอมพิวเตอร์ (คอมมูนิเคชันบอร์ด, สายโคแอ็กเชียล, สายทวีสเต็ดไวร์, ผ่านโมเด็ม, ไม่ผ่านโมเด็ม, ดัมพ์เทอร์มินัล, อินเทลลิเจนต์เทอร์มินัล, อีมูเลเตอร์กับไมโครคอมพิวเตอร์, วิธีรับส่งข้อมูล, ความเร็วในการรับส่ง, ความสามารถ (จำนวนเทอร์มินัล, การขยาย), ราคา, ค่าใช้จ่ายด้านสายสื่อสารเมื่อต่อภายในและเมื่อต่อผ่านโมเด็มออกสู่ภายนอก) ; ซูเปอร์มินิ (ข้อแตกต่างจากมินิคอมพิวเตอร์, ขีดความสามารถเมื่อเทียบกับเมนเฟรม, ราคาเมื่อเทียบกับเมนเฟรม) ; ประวัติความเป็นมาของไมโครคอมพิวเตอร์ (ความต้องการใช้ส่วนตัว, รุ่นและยี่ห้อที่ควรรู้จัก, ราคาในอดีตและปัจจุบัน, วิธีการจัดหา) ; โครงสร้างเครื่องไมโครคอมพิวเตอร์ (เมนบอร์ด, Bus, Slot, Comm1, 2, Disk, Diskette, Keyboard, Display Unit) ; อุปกรณ์เชื่อมโยงเพิ่มเติม (เทปตลับ, จานแม่เหล็กภายนอก, ซีดี, กราฟิกการ์ด, อีเธอร์เน็ตการ์ด, เครื่องพิมพ์ดอต, เครื่องพิมพ์เลเซอร์, เครื่องพิมพ์อิงค์เจ็ต, ซาวด์การ์ด, โมเด็ม) ; ข้อได้เปรียบและเสียเปรียบของไมโครคอมพิวเตอร์ (ความเร็วในการโต้ตอบผ่านคีย์บอร์ดและจอภาพ, ความเร็วในการเข้าถึงข้อมูล, ระบบปฏิบัติการที่ง่ายต่อการใช้งาน, ความเร็วในการพัฒนาโปรแกรม, จอภาพมีแสง, มีเสียง, ข้อจำกัดด้านจำนวนคีย์บอร์ด, ข้อจำกัดด้านขนาดของจานแม่เหล็ก) ; การเชื่อมโยงไมโครคอมพิวเตอร์เป็นเครือข่าย (PC-LAN, Muli-User, สายสื่อสารที่ใช้เชื่อมโยง, ความเร็วในการรับส่งข้อมูล, ราคาและค่าใช้จ่ายของ PC-LAN, ราคาและค่าใช้จ่าย Multi-User) ; การเชื่อมโยงเครื่องคอมพิวเตอร์หลายชนิดเข้าเป็นเครือข่าย (Computer Network, Line Protocol, Protocol Converter, UNIX-LAN, WAN, Standard Protocol (TCP/IP, Ethernet, Token Ring), Router, Hub, สายสื่อสารที่ใช้) ; เครือข่ายมาตรฐาน (Internet, Intranet, DataBase Server, Print Server) ; Micro-Computer in Detail (เรียนการใช้เครื่อง PC) ; Internet in Detail (เรียนการใช้ Internet) ;

2. ความรู้ด้านซิสเต็มส์ซอฟต์แวร์ : ประวัติความเป็นมาซิสเต็มส์ซอฟต์แวร์ (แนวคิดของการมีซิสเต็มส์ซอฟต์แวร์, การเปลี่ยนจาก Bundled เป็น Un-bundled) ; องค์ประกอบหลักของซิสเต็มส์ซอฟต์แวร์ (Operating System, Editor&Compliler, Utilities, File Access Method, Job Control Language, DataBase Language) ; หน้าที่หลักของโอเปอเรติงซิสเต็มส์ (รับคำสั่ง, แสดงผล, แจ้งปัญหา, โหลดและวิ่งโปรแกรม, ดูแลไฟล์และไฟล์ไดเรคทอรี, ตัวกลางในการทำอินพุต-เอาต์พุต, จัดการกับอินเตอร์รัพต์ต่าง ๆ, ควบคุมการใช้เมมโมรี, จัดสรรทรัพยากร) ; การทำงานแบบแบตช์โปรเซสชิง (การจัดงานเข้าทางเครื่องอ่านบัตร, การจัดงานเข้าทางจานแม่เหล็ก) ; การทำงานแบบออนไลน์ (การโหลดและวิ่งโปรแกรมออนไลน์ในเมนเฟรม, การ log-on, การทำ transaction, การ log-off, การหยุดโปรแกรมออนไลน์ในเมนเฟรม, การ log-on และเรียกโปรแกรมมาทำงานในเครื่องมินิคอมพิวเตอร์, การ log-off และเลิกใช้โปรแกรมในเครื่องมินิคอมพิวเตอร์, การ log-on และเรียกใช้โปรแกรมใน PC-LAN, การ log-off และเลิกใช้โปรแกรมใน PC-LAN, การเรียกใช้โปรแกรมในไมโครคอมพิวเตอร์) ; เทคนิคที่ใช้เพิ่มประสิทธิภาพของโอเปอเรติงซิสเต็มส์ (มัลติโปรแกรมมิง, วิชวลสตอเร็จ, รีเอนทรานซ์, เอ็กแซ็กคิวแอนด์สเตเรสซิเดนซ์, มัลติเทรด) ; โอเปอเรติงซิสเต็มส์ที่ควรรู้จัก (ในเมนเฟรม, ในมินิคอมพิวเตอร์, ในไมโครคอมพิวเตอร์, ในเครือข่ายไมโครคอมพิวเตอร์, ในเครือข่ายคอมพิวเตอร์หลายชนิด, ราคา, ค่าบำรุงรักษา) ; ยูทิลิตี้ที่สำคัญ (Sort, merge, Copy, Spooling, Back-up & Recovery) ; ไฟล์แอ็กเซสเมธอด (คำสั่ง READ และ WRITE, Physical Record, Logical Record, Buffer, Alternate Buffer, Sequential Access, Random Access, Indexed-Sequential Method, Indexed Method, Btrive, Relational DataBase) ; จ็อบคอนโทรล (การจัดงานเป็น Job, Job control cards, Job control language) ; Unix in Detail ; Netware in Detail ; Window 95 in Detail ;

3. ความรู้ด้านออกแบบโปรแกรมและเขียนโปรแกรม : ประวัติความเป็นมา (เหตุที่ต้องเขียนโปรแกรม, โปรแกรมแรกของโลก, ภาษาเครื่อง, ภาษาระดับกลาง (Assembly), ภาษาระดับสูง (RPG, FORTRAN, COBOL, ALGOR, PL-1, PASCAL, BASIC, C), ภาษาสำหรับไมโครคอมพิวเตอร์ (BASICA, TURBO PASCAL, C, C++, Visual BASIC, Visual C, Delphi)); ขั้นตอนการสร้างโปรแกรม (Editor, Compiler, Source Program, Object Program, Sub-Program, Linkage, Testing, Debugging, Back-up, Version management) ; โครงสร้างของโปรแกรมแบบโพรซีเจอรัล

หลักสูตรคอมพิวเตอร์กับการพัฒนาทางด้านไอที

(data declaration, sub-routine declaration, main program (imtialization, main loop, exit point, ending)) ; การไหลของโปรแกรม (Flow Chart, algorithm, subroutine, การเรียกใช้ subroutine) ; แนวในการออกแบบโปรแกรม (Top down, Bottom-up, การแยกเป้าหมายใหญ่เป็นเป้าหมายย่อย, การมองหางานที่เหมือน ๆ กันเพื่อทำเป็น subroutine, การมองหาจุดวนลูป); แนวในการเขียนและทดสอบโปรแกรม (การพัฒนาทีละส่วน, การแสดงผลทีละสเตป) ; วิธีการทางโปรแกรมที่ควรทราบ (การทำ editing, การจัดเรียงข้อมูลในเมมโมรี, การจัดเรียงข้อมูลในจานแม่เหล็ก, การค้นหาข้อมูลในเมมโมรี (Sequential search, Binary search), การค้นหาข้อมูลในจานแม่เหล็ก (Sequential search, Binary search, Indexed-Sequential File, การสร้างดัชนีขึ้นมาใช้เอง), การผสมผสานระหว่างเมมโมรีกับจานแม่เหล็กเพื่อเพิ่มประสิทธิภาพของโปรแกรม, การสร้างภาพกราฟิก, การสร้างภาพเคลื่อนที่, การสร้างภาพ 3 มิติ, การหมุนภาพ 3 มิติ) ; การป้องกันปัญหา concurrent update (ปัญหาเกิดขึ้นได้อย่างไร, ต้องแก้ไขด้วยวิธีใด, record locking, read-befor-write technique) ; การป้องกันไฟล์เสียเมื่อไฟฟ้าดับ ; การใช้ภาษาฐานข้อมูล (ภาษาฐานข้อมูลที่ควรรู้จัก, หลักการของระบบฐานข้อมูล, ข้อดีของการใช้ภาษาฐานข้อมูล) ; Turbo Pascal in Detail ; Visual Basic in Detail ; Delphi in Detail ;

4. ความรู้ด้านออกแบบระบบและทำเอกสารประกอบระบบ : ประสบการณ์และปัญหา (ทำโปรแกรมโดยไม่มีเอกสาร, ทำโปรแกรมโดยไม่ดูความต้องการที่แท้จริง, ขาด User participation, เปลี่ยนเป้าหมายไปเรื่อย ๆ, ete) ; การจัดทีม (ประเมินขอบเขตของงาน, ประมาณอัตรากำลัง, กำหนดตัวผู้ประสานงานจากฝ่ายต่าง ๆ (ผู้ใช้, ฝ่ายตรวจสอบ, ทีมคอมพิวเตอร์), วางแผนการทำงาน (ทำ Time Schedule), รีวิว 3 ฝ่าย, มอบหมายงาน) ; การเข้าวิเคราะห์ระบบงานเดิม (ดูโครงสร้างหน่วย, ดูหน้าที่หน่วย, ดูเป้าหมายหน่วย, identify กระบวนการทำงาน (work process) ต่าง ๆ, ดูข้อมูลที่ใช้, ดูแบบฟอร์มที่ใช้, แยกประเภทรายการ, สัมภาษณ์หัวหน้าหน่วยงาน, สัมภาษณ์ผู้บันทึกข้อมูล, สัมภาษณ์ผู้ใช้ข้อมูล, นับปริมาณงาน, นับปริมาณข้อมูล, รวบรวมแบบฟอร์มและรายงาน) ; การออกแบบระบบงานในด้านผู้ใช้ (จัดฐานข้อมูลอย่างคร่าว ๆ, จัดกระบวนการทำงานใหม่, จัดวิธีการควบคุม, ประมาณแรงงาน, ประมาณเวลาทำงานแต่ละ Transaction, ร่าง User Manual, นำเสนอผู้บริหารของผู้ใช้) ; การออกแบบระบบด้านคอมพิวเตอร์ (ออกแบบโครงสร้างฐานข้อมูล, กำหนด data element ที่จะเก็บ, กำหนดรูปร่างของเรคอร์ด, กำหนดการทำ editing, แยกเป็นโปรแกรมต่าง ๆ, กำหนดฟังก์ชันของโปรแกรมต่าง ๆ, กำหนดขั้นตอนการใช้งานของโปรแกรมต่าง ๆ, ร่าง Program Spec, ทำ system walk-thru (ดูความสอดคล้องของโปรแกรม, ดูว่าจะแก้ปัญหาต่าง ๆ ได้ครบถ้วนหรือไม่), ประมาณการใช้ resource, ประมาณ Response time, ประมาณค่าใช้จ่าย, รีวิว cost-benefit, ร่าง System Manaul) ; การรีวิวระบบกับยูสเซอร์ก่อนลงมือเขียนโปรแกรม (ปรับ User Manual, ปรับ System Manual, รีวิว cost-benefit, นำเสนอเพื่อขออนุมัติการลงมือเขียนโปรแกรม) ; เทคนิคการเลือกวิธีการทำโปรเซสชิง (On-line, Off-line, Real-time, Batch Processing, ลักษณะงาน (front office, back office), ลักษณะการใช้ข้อมูล, ความเร่งด่วน) ; เทคนิคการประมาณการใช้ Resource (ขนาดของ file, ขนาดของโปรแกรม, work area, CPU Utilization, Printer Speed, ขนาดเมมโมรี, ชนิดของ back-up, จำนวน generation ของ back-up, จำนวนม้วนเทป, จำนวนตลับเทป, จำนวนดิสก์เกตต์, จำนวนกระดาษ, จำนวนคนปฏิบัติงาน) ; เทคนิคการประมาณ Response Time (จำนวน Transaction ต่อวัน, Peak Period, จำนวน instruction ที่ต้องทำต่อ 1 transacion, จำนวน I/O ที่ต้องทำต่อ 1 Transaction, Transmission time, CPU Utilization, Disk Utilization, Line Utilization, จำนวน character ที่ต้องคีย์, จำนวน character ที่ต้องส่งไปมา) ; เทคนิคการทำ Cost-benefit Analysis (Onetime cost, Runing cost, การคำนวณเป็น Present Value, Tangible benefit, In-tangible benefit, การประมาณ In-tangible benefit เป็นตัวเงิน) ; Inventory Control in Detail;

5. ความรู้ด้านการควบคุมโครงการและติดตั้งใช้งาน : ประสบการณ์และปัญหา (ขาดผู้ควบคุมโครงการ, ไม่รู้ว่าใครมีอำนาจสูงสุด, ไม่รู้วิธีประสานงาน, ยูสเซอร์ไม่รู้เรื่องคอมพิวเตอร์, คนคอมพิวเตอร์ไม่รู้งานของยูสเซอร์, ไม่มีแผน, ประมาณเวลาและแรงงานผิด, ผู้บริหารไม่ให้แรงสนับสนุน, ขาดความรู้ในการควบคุมโครงการ) ; System Life Cycle (Feasibility Study, System Analysis, Business System Design, Data Processing Design, Programming, Testing, Installation, Evaluation) ; การจัดทีม (user group, computer group, auditor group) ; หน้าที่และความรับผิดชอบของผู้ควบคุมโครงการ (ติดตามงาน, จัดประชุมรายงานความก้าวหน้า, จัดประชุม

หลักสูตรคอมพิวเตอร์กับการพัฒนาทางด้านไอที

แก้ปัญหา, ดูแลการทำบันทึกและเอกสาร, ประเมินสถานการณ์, re-schedule เมื่อจำเป็น, re-assign เมื่อจำเป็น, ควบคุมค่าใช้จ่าย, ควบคุมการใช้งาน resource, รายงานผู้บริหาร) ; หน้าที่ความรับผิดชอบของ User group (กำหนดความต้องการ, ให้ข้อมูล, ร่วมออกแบบด้าน Business, ร่วมทดสอบระบบ, ตรวจรับ, คัดเลือกพนักงานลงตำแหน่งใหม่, จัดฝึกอบรม, ร่วมประเมินผล) ; หน้าที่และความรับผิดชอบของ Computer group (ทำ Feasibilty Study, ร่วมออกแบบด้าน Business, ออกแบบด้าน Data Processing, พัฒนาโปรแกรม, ทดสอบแต่ละโปแกรม, ร่วมทดสอบระบบ, ทำเอกสาร (System manual, User manual, Program Spec), ร่วมประเมินผล, ปรับแก้ระบบ) ; หน้าที่และความรับผิดชอบของ Auditor group (พิจารณาว่าจะร่วมในขั้นตอนใดบ้าง, ต้องร่วมในการทดสอบระบบ, ต้องร่วมในการประเมินผล) ; การควบคุมเอกสาร (กำหนดวิธีการให้หมายเลขกำกับบันทึกและเอกสาร, ดูแลให้เป็นไปตามกำหนด) ; การควบคุมการประชุม (เปิดประชุม, กำหนดเป้า, ผลักดันให้ทุกคนออกความคิดเห็น, จับประเด็น, รักษาความเรียบร้อย, หลีกเลี่ยงการเผชิญหน้า, บำรุงขวัญ, หาข้อสรุป) ; การติดตามงานในช่วงโปรแกรมมิง (กำหนดให้โปรแกรมเมอร์ทุกคนรายงานความคืบหน้า, ดูการใช้เวลาและแรงงาน, ดูความคืบหน้า, ดูการใช้ resource, มองล่วงหน้าเพื่อป้องกันความล่าช้าของคนหนึ่งไปทำให้คนอื่นล่าช้าด้วย, ให้คำปรึกษากับผู้ล่าช้า, re-assign เมื่อจำเป็น) ; การติดตามงานในช่วงทดสอบระบบ (ให้หมายเลขแก่ข้อผิดพลาดที่ตรวจพบ, ติดตามการแก้ปัญหาแต่ละข้อที่ตรวจพบ, ดูจำนวนข้อผิดพลาดที่เหลืออยู่, ติดตามและผลักดันให้แก้ปัญหา) ; การควบคุมการใช้ resource (กำหนดตารางการใช้เครื่องมือเครื่องจักร, แก้ปัญหา conflict ระหว่างทีม, ติดตามการจัดซื้อจัดหามาเพิ่มเติม) ; การติดตามงานในช่วง Installation (ให้ยูสเซอร์คัดเลือกพนักงานลงตำแหน่งใหม่, จัดให้มีการฝึกอบรม, บังคับให้ทำเอกสารประกอบการฝึกอบรม, ประเมินผลการฝึกอบรม) ; การรายงานต่อผู้บริหาร (รายงานด้วยบันทึกทุกเดือน, รายงานด้วยจาวาทุกครั้งที่ขั้นตอนสำคัญ ๆ เสร็จ, รายงานด้วยจาวาทุกครั้งที่มีปัญหากระทบต่องบประมาณ, บันทึกขออนุมัติการ cut over) ;

6. ทบทวนความรู้ด้านฟิสิกส์และไฟฟ้า : สารตัวนำไฟฟ้า ; สารกึ่งตัวนำ ; ไฟกระแสตรง ; ไฟกระแสสลับ ; หน่วยวัดแรงดันไฟฟ้า (Volt, Millivolt, Kilovolt) ; หน่วยวัดกระแสไฟฟ้า (Ampare, Milliampare, Microampare, Kiloampare) ; สนามแม่เหล็ก ; หน่วยวัดสนามแม่เหล็ก (Guass) ; การเหนี่ยวนำบนเส้นลวดที่ตัดผ่านสนามแม่เหล็ก ; การเหนี่ยวนำแม่เหล็กเมื่อใส่กระแสไฟผ่านขดลวด ; วงจรไฟฟ้าทั่วไป (ต้นกำเนิดแรงดัน, ตัวต้านทานแบบ Resistive, ตัวต้านทานแบบ Inductive, ตัวต้านทานแบบ Capacitive, ความสัมพันธ์ระหว่างแรงดันกับกระแสกับความถี่, เครื่องขยายเสียง, เครื่องรับวิทยุ, เครื่องส่งวิทยุ) ; ทรานซิสเตอร์ ; ไดโอด ; วงจรไฟฟ้าที่มีทรานซิสเตอร์และไดโอด (เครื่องขยายเสียง, อิเล็กทรอนิกส์สวิตช์ (AND-gate, OR-gate, NOR-gate), วงจร Bistable, Binary Counter, Shift Register, วงจรรวมขนาดเล็ก, วงจรรวมขนาดใหญ่) ; คลื่นวิทยุ (Wave length, ความถี่, การกระจาย, การดูดซับ) ; หน่วยวัดความถี่คลื่นวิทยุ (Hertz, Kilohertz, Megahertz, Gigahertz) ; การเคลื่อนที่ของคลื่นบนสายสื่อสาร ; Square Wave ; การแตก square wave เป็นคลื่นวิทยุ ; การใช้ square wave เป็นรหัส binary ; การรับส่งข้อมูลบนสายสื่อสารแบบต่าง ๆ (Coax, Twisted pair, Telephone line, fiber opties, การดูดซับคลื่น, distortion, time multiplexing) ; แม่เหล็ก, สารแม่เหล็ก ; การเหนี่ยวนำแม่เหล็ก ; วงแหวนแม่เหล็ก ; การบันทึกข้อมูลด้วยวงแหวนแม่เหล็ก ; การบันทึกข้อมูลบนจานแม่เหล็ก ; หลอดสูญญากาศ ; การไหลของอิเล็กตรอนในสูญญากาศ ; การ excite อะตอมด้วยการชนของอิเล็กตรอน ; การปลดปล่อยพลังงานของ excited electrox เมื่อกลับที่เดิม ; แสงเลเซอร์ ; วัตถุเรืองแสง ; หลอดคาโทด ; หลักการทำงานของจอภาพ ; หลักการทำงานของจอภาพสี ;

7. ทบทวนความรู้ด้านคณิตศาสตร์และตรรกวิทยา : ชนิดของตัวเลข (Intiger, Floating Point หรือ Real, ค่าอนันต์,....) ; เลขฐานสิบ (การนับ, การแสดงค่าด้วยภาพตัวเลข, mantiss, exponent, จุดทศนิยม, การบวก-ลบ-คูณ-หาร, ขั้นตอนการทำงาน, algorithm, การหาค่ารูท 2) ; เลขฐานสอง (การนับ, การแสดงด้วยภาพตัวเลข, mantiss, Exponent, เลขยกกำลังที่ติดลบ, การบวก-ลบ-คูณ-หาร) ; การเปลี่ยนฐานระหว่างฐานสิบกับฐานสอง ; Boolean Algebra (Set, Union ของ Set, Intersection ของ Set, Boolean Operator ต่าง ๆ (AND, OR, NOT, XOR, NAND,...), Boolean Expression, การยุบและแยก Boolean Expression, การประยุกต์กับงานสร้างวงจรไฟฟ้า (Binary Adder, BCD Adder, Shift Register...)) ; Numerical Methods (series, convergent series, divergent series, successive approximation, Newton's method, Polynomial Approximation,

หลักสูตรคอมพิวเตอร์กับการพัฒนาทางด้านไอที

Summation, Random Number, Monte Carlo Methods) ; Algebra (function, equation, simultaneous equations, Matrix, Matrix inversion) ; Permutation & Combination (outcom, number of outcomes, Probability of outcome, การโยนเหรียญ, การโยนลูกเต๋า, การเล่นไพ่) ;

8. ทบทวนความรู้ด้านสถิติและบริหารธุรกิจ : การใช้ข้อมูลอย่างง่าย (การเก็บข้อมูล, การวิเคราะห์ข้อมูล, การแจงนับข้อมูล, ตารางความถี่ข้อมูล, การให้รหัสข้อมูล, probability, การใช้ประโยชน์จากตารางความถี่) ; ทฤษฎีทางสถิติ (Law of Large Number, Normal Distribution, statistical Inference, Regression, Time Series, Least squares Method, Multiple Regression, Variance Analysis) ; การเก็บสถิติ (sampling size, sampling method, การคำนวณความน่าเชื่อถือ) ; การตัดสินใจ (การตัดสินใจแบบฉับพลัน, การตัดสินใจด้วยการวิเคราะห์ข้อมูลอย่างคร่าว ๆ (ดูข้อดีข้อเสียเปรียบเทียบกัน = Qualitative Decission making), การตัดสินใจด้วยการวิเคราะห์ข้อมูลอย่างลึก (Modeling, Forecasting, Simulation, cost-benefit analysis = Quantitative Decission Making)) ; การปรับปรุงบริการ (Time and Motion Study, Queing Theory (distribution of arrival, distribution of service time, distribution of wait time, simulation of queque), การจัดคิวแบบคิวเดียวแต่หลายจุดบริการ, การคำนวณ Utilization factor ของอุปกรณ์ต่าง ๆ) ;

โอเค นี้คือหลักสูตรสร้าง "นักคอมพิวเตอร์" แบบของผม ในหัวข้อวิชาที่ 1. ถึง 5. นั้น ตอนท้ายจะมีคำว่า "In Detail" อยู่ 2-3 เรื่อง เช่น Micro-Computer in Detail, Internet in Detail, Unix in Detail, Netware in Detail, ... เหล่านี้ผมหมายถึงการเรียนในรายละเอียดและให้มีการลงมือทำกันจริง ๆ โดยเฉพาะอย่างยิ่ง ในการใช้ภาษาโปรแกรม 3 อย่างคือ Turbo Pascal, Visual Basic และ Delphi นั้นต้องทำการบ้านไม่ต่ำกว่าภาษาละ 5 โปรแกรม ทั้งนี้โดยให้เป็นโปรแกรมคำนวณอย่างง่าย ๆ 1 โปรแกรม โปรแกรมคำนวณและสร้างแฟ้มข้อมูลอย่างง่าย 1 โปรแกรม โปรแกรมคำนวณและสร้างแฟ้มข้อมูลอย่างยาก (ทำ index file เอง) 1 โปรแกรม โปรแกรมที่เน้นการทำอีดิตติงของข้อมูลรับเข้าทางคีย์บอร์ด 1 โปรแกรม และโปรแกรมคอมพิวเตอร์ 1 โปรแกรม ในทุก ๆ โปรแกรมที่ทำขึ้นมานี้จะต้องให้นักศึกษาประมาณความเร็วในการทำงาน (transaction per second, response time) เอาไว้ล่วงหน้า แล้วเช็กกับผลงานจริงเมื่อโปรแกรมทำเสร็จแล้ว

ครับ หลักสูตรทั้งหมดนี้คือสิ่งที่ผมพอจะคิดได้ในเวลา 3 ชั่วโมง ผมเชื่อว่ามันยังจะต้องนำไปปรับปรุงหรือเพิ่มเติมในรายละเอียดอีกมาก หลาย ๆ ท่านอ่านดูแล้วอาจบอกว่ามันเยอะแยะและเยิ่นเย้อเกินไป แต่ผมบอกได้ว่าทุกหัวข้อย่อยนั้นมีความจำเป็นทั้งนั้น ถ้าท่านไม่รู้จัก Normal Distribution หรือตัว Bell Shape ท่านก็คงจะมองภาพของการเกิดข้อมูลรอบ ๆ ตัวท่านไม่ได้ เช่น ไม่รู้ว่าคนไทยเรานั้นเกิดมาแล้วจะมีความสูงแตกต่างกันมากน้อยแค่ไหน และคนส่วนใหญ่จะมีความสูงสักเท่าใด คนที่สูงมาก ๆ และคนที่ต่ำเตี้ยมาก ๆ นั้นทำไมจึงมีน้อย และเมื่อจะผลิตเสื้อผ้าออกมาขายจะผลิตสำหรับคนความสูงระดับต่าง ๆ ในจำนวนเท่าใด ถ้าไปคิดง่าย ๆ ว่าผลิตทุกขนาดเท่า ๆ กัน ท่านก็คงจะขาดทุนแน่นอน ถ้าท่านไม่มีความคิดหรือความเข้าใจเหล่านี้ ท่านก็คงจะไปออกแบบระบบงานไม่ได้ ท่านจะลืมถามหรือลืมหาข้อมูลเพื่อดูการกระจายความถี่เหล่านี้ แล้วระบบงานของท่านก็จะไม่เข้ากับโลกภายนอก

ครับ ผมหวังว่าบทความนี้คงพอจะให้แนวทางในการพัฒนานักคอมพิวเตอร์ที่เหมาะสมให้แก่ประเทศได้ และผมขอย้ำอีกครั้งหนึ่งว่า ถ้าคิดจะรับจากคนตกงานในยุค IMF นี้ ก็ขอให้ทดสอบพื้นความรู้ในข้อ 6., 7., 8. เสียก่อนด้วย

เอ้อ เกือบลืมเรื่องที่สำคัญไปอย่างหนึ่ง คือแต่ละวิชาควรใช้เวลาประมาณสักเท่าใด เรื่องนี้ผมพอประมาณคร่าว ๆ ได้ดังตารางที่ 1

ตารางที่ 1

วิชา	ชั่วโมงเรียน	ชั่วโมงห้องทดลอง
1. ความรู้ด้านฮาร์ดแวร์	240	120
2. ความรู้ด้านซิสเต็มส์ซอฟต์แวร์	240	120
3. ความรู้ด้านออกแบบโปรแกรมและเขียนฯ	240	240
4. ความรู้ด้านออกแบบระบบและฯ	120	60
5. ความรู้ด้านควบคุมโครงการและฯ	120	60
6. ทบทวนความรู้ด้านฟิสิกส์และไฟฟ้า	40	-
7. ทบทวนและความรู้ด้านคณิตศาสตร์และตรรก	40	-
8. ทบทวนความรู้ด้านสถิติและบริหาร	40	-